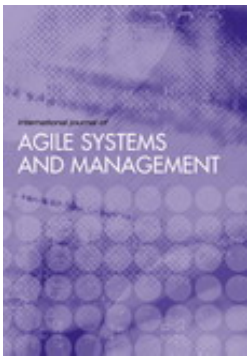


International Journal of Agile Systems and Management

ISSN online: 1741-9182 - ISSN print: 1741-9174
https://www.inderscience.com/ijasm

Logics alignment in agile software design processes

Diana Chronéer, Mari Runardotter, Jeaneth Johansson

DOI: 10.1504/IJASM.2024.10067040

Article History:

Received:	27 August 2024
Last revised:	05 September 2024
Accepted:	07 September 2024
Published online:	08 October 2024

Logics alignment in agile software design processes

Diana Chronéer* and Mari Runardotter

Information Systems,
Luleå University of Technology,
971 87 Luleå, Sweden
Email: Diana.Chroneer@ltu.se
Email: mari.runardotter@ltu.se
*Corresponding author

Jeaneth Johansson

Entrepreneurship & Innovation,
Luleå University of Technology,
971 87 Luleå, Sweden

and

School of Business,
Innovation and Sustainability,
Halmstad University,
Kristian IV:s väg, 3301 18 Halmstad, Sweden
Email: jeaneth.johansson@ltu.se

Abstract: We propose that technological, service-dominant and design logics must interplay for an IT artefact to succeed. Based on data from a project aiming at a B2B platform for manufacturing small and medium enterprises (SMEs) in Europe, we explore these three logics in an agile software design context. By using an inductive approach, we theorise about what is needed for the alignment of the three logics. We contribute with a novel theoretical lens, the Framework for Adaptive Space. We offer insights into the importance of continuously reflecting on all three logics during the agile software design process to ensure mutual understanding among the agile team and the B2B platform end-users involved.

Keywords: design thinking; agile software design; adaptive capabilities; technological logic; service-dominant logic; design logic.

Reference to this paper should be made as follows: Chronéer, D., Runardotter, M. and Johansson, J. (2024) 'Logics alignment in agile software design processes', *Int. J. Agile Systems and Management*, Vol. 17, No. 6, pp.1–31.

Biographical notes: Diana Chronéer, Associate Professor, PhD in Industrial Management, Luleå University of Technology, but is currently working in the research group Information Systems. Her main research interests are in digital transformation and service innovation, business model development, information logistics, and project management. Project management relates to value creation and learning in project and research related to business model focus on how to develop appropriate strategies for business model development and collaboration models in digital settings.

Mari Runardotter, Associate Professor, PhD in Social Informatics, Luleå University of Technology. Her research concerns possibilities, consequences, and effects of IT for humans, organisations, and society. Currently, her interest is focused on the ongoing digitalisation and digital transformation of society. Additional research areas are open innovation, and e-government/ e-governance. She also build on (critical) systems thinking, considering how things, phenomena, and effects etc. are interrelated and influences each other. Of interest is also ethics, gender, and diversity aspects. To involve end-users in various digitalisation/digital transformation initiatives require innovative methods, processes, and tools. She has genuine experience of user participation and involvement, and methods to support this. She is part of Botnia Living Lab, run by the research group Responsible Research and Innovation at LTU.

Jeaneth Johansson, Professor, PhD in Accounting and Control, Luleå University of Technology but is also associated to Halmstad University. She has extended experiences of entrepreneurship and innovation research. Her expertise covers financial decision-making, business model innovation as well as commercialisation and implementation of new innovative solutions. Research is based on theoretical grounded methodologies, close to practice involving guidance of business development. She has published in journals such as *Entrepreneurship Theory and Practice, Harvard Business Review, Small Business Economics, Technological Forecasting and Social Change and Journal of Management Studies*.

1 Introduction

The relation between design thinking and agile software design is observed by many scholars (cf. Pereira and Russo, 2018; Wangsa et al., 2022). While design thinking is a human-centred problem-solving approach, focusing on analysis, synthesis and generation of insights and ideas for the benefit of end-users, agile software design concentrates on the rapid and adaptive design of artefacts in an effective design process that produces value for users (Wangsa et al., 2022). That is, in agile software design, the ability to adaptively promote a quick response to changes in the technological artefact is emphasised (Al-Saqqa et al., 2020). As such, design thinking rests on a design logic that emphasises user needs, wants and requirements, while the technological logic is at focus in the agile software design, where the artefact is iteratively developed by teamwork in collaborative processes that should lead to customer value (Wangsa et al., 2022). Further, the intended users bring in their logic, which relates to business value, hence, a service-dominant logic. We refer to logic as the system or principles of reasoning applicable to any branch of knowledge (cf. https://dictionary.cambridge.org). Thus, we use the term 'logics' to indicate that all those involved in the design of a B2B platform bring their knowledge, their experience, and their interpretation of what is value-creating and important. During the agile software design process, they lean on this when evaluating and assessing the progress of the work (from needs and requirements to the final solution), which often means that what is obvious to one group contrasts with that of another.

This paper is built on a project aiming at a B2B platform for manufacturing SMEs in Europe. Today, B2B platforms act as expansive marketplaces, enabling rapid commercialisation and scaling of business and innovation (Kamalaldin et al., 2020;

Burström et al., 2021; Pauli et al., 2021; Palmié et al., 2022). Platforms have two central roles they can perform. From a technological perspective, platforms enable the creation of complementary solutions by others, and from a market intermediary perspective, platforms facilitate transactions and interactions between different groups of actors (e.g., sellers and buyers) by offering a marketplace (Pauli et al., 2021). B2B platforms often jointly play both of these roles. As such, B2B platforms enable distributed economic and social actors to exchange and integrate resources for mutual value creation (Blaschke et al., 2019), and are an important building block for Industry 4.0 (Pauli et al., 2021). Despite the growing interest and potential impact of these platforms, many struggle to unlock value and seize opportunities (Pidun et al., 2020). A collaborative B2B context is characterised by organisational complexity with many different types of actors with different roles, needs, and incentives (leading to different logics). Companies grapple with the challenge of aligning incentives for the different actors within evolving platform ecosystems, missing the target of customer acceptance (Hauke-Lopes et al., 2022). Furthermore, failure is also often explained by technological challenges due to complexity, lack of openness and standardisation (Tessmann and Elbert, 2022) requiring integration (Hein et al., 2019). One of the biggest digital platform failures is simply not understanding the need to design for interaction, engagement, and the business value of the platform (Van Alstyne et al., 2016). In addition, another critical reason for failures is often connected to the lack of usability. The literature recognises that usability methods are often used too late in the development process (Brhel et al., 2015). We agree with Van Alstyne et al. (2016) that there is a need to transform ways of thinking when designing digital systems, i.e., to identify, understand and integrate different logics in design processes.

Current information systems literature on agile research mainly covers software development projects (Paasivaara et al., 2018), and the agile perspective focuses especially on challenges in large agile software design projects, e.g., inter-team coordination, customer engagement, and agile requirements design (Mamonov and Peterson, 2021). Fundamental factors in agile methods are related to human factors, e.g., an agile environment must enable system designers to work efficiently, collaborate, and share skills. However, these human factors are often not fully understood by agile practitioners (Sampaio et al., 2021). This has led to a gap between what companies are expecting and what the system designers really know, besides technical knowledge, in terms of soft skills such as communication, teamwork, and leadership (Sampaio et al., 2021). Further, a problem is that agile practitioners usually focus on delivering the developed software but neglect the user experience, and hardly discuss users' context (Curcio et al., 2019). Hence, in line with Curcio et al. (2019), we argue that there is a need for a better understanding and integration between the different logics that different actors hold in an agile software design process, to make the agile software design process more effective.

In the context of a project aiming at developing a B2B platform, we focus on systems designers' practice since they, regardless of how agile the design process is, need to communicate changes, decide things in run-time, etc. There are several terms for those who design and develop IT systems and software, such as software engineer, architecture, and software designer/developer, and system designer. We will use the term system designer as a generic term. Design embodies a spectrum of practices undertaken by professional designers and those who actively engage in their pursuits (Kimbell, 2011, 2012). Karpen et al. (2017) assert that design processes need to be grounded in principles

that emphasise design's inherently human- and meaning-cantered nature, as well as its co-creative, inclusive, holistic, and contextual nature. Therefore, our focal point revolves around three different logics in agile software design, crucial for crafting design solutions that cater to the B2B-platform customers' and users' needs and desires (cf. Carlsson et al., 2011). Accordingly, our aim is to *explore the technical, the service-dominant, and the design logics involved in the agile software design process of a B2B platform, as well as how these logics interplay in agile software design processes*.

The paper makes two significant contributions, collectively advancing our comprehension of agile software design for B2B platforms. First, in terms of theoretical advancements, our study outlines a comprehensive framework of adaptive space in software design processes. Second, the paper provides practical insights by making the three logics operating in agile software design transparent for navigating the operational intricacies of B2B platform design within system design projects.

In the upcoming sections, the paper unveils its narrative. We start by mapping the theoretical landscape, shedding light on design thinking in relation to agile software design and service-dominant logic. Then we introduce the research method, presenting the Gioia methodology used for elucidating the challenges in agile software design processes, before shifting towards our empirically identified dimensions and their relevance for an alignment of logics within system design projects. Synthesising our insights, we unveil a framework for adaptive space, which highlights the importance of recurrent reflection of the different logics involved.

2 Software design for B2B platforms

B2B-platform systems are designed for heterogeneous actors (Bonina et al., 2021; Hanseth and Lyytinen, 2010; Spagnoletti et al., 2015) with different needs and wants. This carries implications for the design processes and poses challenges for system designers to consider heterogeneous user needs in their platform designs (de Reuver et al., 2018; Evans and Schmalensee, 2016; Islind et al., 2019). Furthermore, creating value on platforms involves design choices related to the roles of these actors using the platform and ensuring feedback of them from the beginning of the design process (Tura et al., 2018).

In software design development for B2B-platforms, Agile methods as well as User Experience (UX) design methods aim to build quality software, though, each method approaches development from a different perspective (cf. Ferreira et al., 2011). However, studies show that integration of the two methods has been lacking in software development (Hinderks et al., 2022). Though, system designers need to start by thinking through the issues related to platform launch and diffusion, how to attract participants, and how to ensure easy access to the platform (Tura et al., 2018; Van Alstyne et al., 2016). Therefore, studies point on the need for system designers to understand who the actors are, their different roles, what creates value for those different actors, and how to achieve commitment (Spagnoletti et al., 2015; Tura et al., 2018). Hence, it is central to consider underpinning value co-creation processes, and the characteristics of the business environments (Blaschke et al., 2019) since this will guide and determine the setup of the architecture design of the platform, and thereby also impact the development of collaboration models, and business models. As such, different logics must be integrated into the design process to respond to users' needs, wants and requirements.

2.1 Agile software design

Since B2B-platform systems are designed for heterogeneous actors, the system design activity becomes complex due to the need for direct user participation in software modelling to generate a solution in an integrated manner (Pereira and Russo, 2018). Agile software design and methods provide an iterative way to make effective and efficient system development that provide opportunities for team members to increase communication (Piedrahita et al., 2023), enables acceleration of software delivery, manages prioritisation changes, and increases productivity (Ghayyur et al., 2018; Pereira and Russo, 2018; Motwani and Kataria, 2023).

The agile software design approach was proposed to provide flexibility and to adopt changes in requirements at any stage through a certain set of values and principles (Al-Saqqa et al., 2020), described in the "Manifesto for Agile Software Development" (Agile manifesto). By reducing the time between the design and deployment, agile methods allow software to be released to users more quickly (Pereira and Russo, 2018). The advantages of agile-based software design are several. Erickson et al. (2005) argue that it improves communication and coordination among team members; leads to quick deliveries and releases; facilitates design flexibility and leads to a more reasonable and flexible process. Additionally, it aims to increase value to customer collaboration, individuals, and interactions (Curcio et al., 2019). It also has a significantly positive impact on software development performance, quality, and user satisfaction (Fitriani et al., 2016). Satria et al. (2017) argue that agile software design helps manage time and requirements from the users' perspective, and that the never-ending requirements of the user can be handled. However, Bannerman and Thorogood (2012) emphasise that it is the usability of the artefact, and not the scope of the project, that indicates the success of an agile project.

Agile methods also have disadvantages. Satria et al. (2017) mention the following: too focused on the functionality of the system; management of project resources (codes, documentation, knowledge) emerge in the development process; and the focus is on software functionality, not software usability. Another shortcoming is that system development documentation is poor, which hinders knowledge transfer since much communication happens verbally and informally (Theunissen et al., 2022). Hence, agile methods might not be the best option for extensively large groups of issues in managing face-to-face communications, and setup complexities (Wangsa et al., 2022). This is a drawback since coordination is one of the biggest challenges associated with large-scale software development today (Gustavsson, 2023).

Agile methods have become the norm in system development today (Gustavsson, 2023), and several methods and frameworks for agile software design have emerged based on the Agile Manifesto, e.g., test-driven development (TDD), feature driven development (FDD), extreme programming (XP), Scrum, and dynamic system development model (DSDM) etc. (Erickson et al., 2005; Campanelli and Parreiras, 2015; Alqudah and Razali, 2016). Further, there are studies highlighting that gamification seems to have the potential to improve the adoption of agile practices (Marques et al., 2023). Although agile methods and practices require close collaboration with customers, the focus is often on how useful software can be developed, and customer value is primarily driven by providing an appropriate functional scope (Fitzgerald, 2012), not focusing on actual usability, i.e., "the extent to which a software can be used by specified

users to achieve specified goals effectively, efficiently, and satisfactorily in a specified use context" (Brhel et al., 2015, p.2).

Agile software design teams should be built up with different roles and competencies involved, such as technically focused system designers, user interface and interaction or user experience (UX) designers (cf. Lehnen et al., 2016). However, Curcio et al. (2019) states that in some agile methodologies, no defined role represents a user interface specialist or an interaction designer, and hence, the role in agile teams is not always clear. For instance, the project owner is the 'voice of the customer' but is also intended to represent all the stakeholders who have input into the product's requirements. However, can a project owner represent different end-users of a system? End user is a generic term, and a customer might be one instance of an end user but might not be the only type of end user (Lehnen et al., 2016). Highlighting a design thinking approach can promote communication between the agile software design teams and customers throughout the entire software development project (Pereira and Russo, 2018), and hence the understanding of the different users. Both design thinking and agile approaches offer interaction and collaborative opportunities to better understand the problem. Design thinking applies human-centred design as a key to addressing problem-solving needs, while agile focuses quickly on problem identification and resolution within a short time frame (Wangsa et al., 2022). Thus, human factors are fundamental, and agile practices also include human and social aspects (Hoda et al., 2018). How the developer teams communicate in the project is vital, however, the other teams sometimes leave out the UX team, since agile methods are more focused on how to deliver functionality than on user experience and usability.

Studies highlight the importance to integrate UX in Agile software development (cf. Ferreira et al., 2011; Hinderks et al., 2022; Persson et al., 2022). In a review by Hinderks et al. (2022), they highlight that no papers consider both managing the UX process and Agile software development. For instance, upfront user-centred design (UCD) is a frequent used approach, though as Hinderks et al. (2022) point out it does not integrate UX methods in Agile software development, it only coordinates the work of teams. A study by Persson et al. (2022) show how integration through adjustments made distinct contributions to UX designers' and software developers' pursuit of agility. However, Persson et al. (2022) state that the integration of UX activities with agile development is not easy. Also, Pereira et al. (2024) highlight that many teams are today virtual, and coordination, planning and project management need further research to better fit the needs of these virtual teams.

Thus, Hinderks et al. (2022) mean that using a single UX method is not the solution, and that future research should focus more on the integration of UX and Agile. Further research is needed on the relevance of soft skills for agile systems designers (Sampaio et al., 2021) to improve agile software design (Satria et al., 2017). Brhel et al. (2015) propose that future research should focus on identifying further principles in the people/social as well as the technological dimension of user-centred agile software design. Curcio et al. (2019) suggest more research on the integration of usability in agile software design through tools and people (teams). Pereira and Russo (2018) highlight a need for research that can present rigorous empirical findings about adopting design thinking integrated into agile software design. However, there are attempts to bridge the gaps between design thinking and agile. For instance, Google employees developed an approach labelled 'the design sprint', borrowing some techniques from design thinking, which has shortened the time to establish the solution (Wangsa et al., 2022).

2.2 Design thinking

When designing software for a B2B-platform, system designers, and interaction/UX designers, together with different internal and external actors, i.e., a multi-disciplinary team, explore the interplay of problem and solution space and refine decisions that shape the final design and behaviour of the software product. Highlighted by Persson et al. (2022), UX professionals have many different roles, e.g., user experience designer, UX analyst, UX evaluator, UX manager. UX designers, who specialise in interaction design, but struggle to integrate their work into agile processes (Persson et al., 2022).

Throughout the design process, system designers with various roles jointly design the system, i.e., not only do they elicit requirements, but they also design the requirements by discussing and shaping these requirements with the contributing stakeholders; design code by modularising, composing, analysing, and evaluating source code; and they design use cases, and user interfaces (Jolak et al., 2023). Since these activities require creativity and handling complex problems, Jolak et al. (2023) and Canedo et al. (2020) promote design thinking. Since design thinking is a people-centred model that is focused on creativity and innovation to create a product or service that resolves a complex problem for the end-users, it supports multi-disciplinary teams' collaboration (Nedeltcheva and Shoikova, 2017).

Originally conceived as an innovation process (Brown, 2008; Brown and Wyatt, 2010), design thinking has also proved to be a problem-solving process that is possible to apply to every context that strives for innovation or improvement (Liedtka, 2014; Dell'Era et al., 2020), i.e., also design of "new sorts of processes, services, interactions, entertainment forms, and ways of communicating and collaborating" (Brown and Katz, 2011, p.381). As such, design thinking also attracts the management and business world (Dell'Era et al., 2020), and has been further developed by combining with, e.g., social practice research (Hoolohan and Browne, 2020), and constructivism (Pande and Bharathi, 2020). Parizi et al. (2022) argue that the iterative approach to problem-solving that characterises design thinking has led to its integration with agile methods.

In essence, design thinking is an approach that emphasises user involvement and to understand the people for whom products or services are (to be) designed (Dam and Siang, 2018). The focus is on identifying human needs and using rapid prototyping and interactive learning cycles early in product, service, and system development processes (Pereira et al., 2021). Multidisciplinary teams that jointly explore technologies and processes aimed at meeting users' expectations are essential (Parizi et al., 2022). Reaching agreement on what matters to users allows design teams to achieve focus and create a positive impact (Liedtka, 2014). Therefore, design thinking is characterised by emphasising discovery over solution generation, expanding the boundaries of both problem definitions and solutions, and emphasising co-creation (Liedtka et al., 2013). According to Pereira et al. (2021), the benefits of design thinking are that collaboration, understanding, empathy, and creativity increase, the software solution is assertively defined, and real needs are identified in an efficient process. However, design thinking is also accompanied by challenges, such as lack of time, commitment, value, and knowledge, and difficulties in converging all stakeholders' insights (Pereira et al., 2021).

Kimbell (2011) asserts that the concept of 'design thinking' should denote something more than design and question what 'thinking' implies. Making a distinction between 'thinking' and 'doing' and between designers and the world for which they design, neglects design thinking in practice, argues Kimbell (2011) and advocates a view of

design that is "a situated, contingent set of practices carried out by professional designers and those who participate in designers' activities" (Kimbell, 2011, p.286). The socio-material world is created through practice; thus, we should view design activities as distributed across several different people and artefacts designed together (Kimbell, 2011). Doing design as practice acknowledges that design practices are "habitual, possibly rule-governed, often routinised, conscious or unconscious, and that they are embodied and situated" (Kimbell, 2012, p.135). Doing design requires creativity, which in turn links to uncertainties throughout the project (Kimbell, 2009). Accordingly, designers can never be sure that their design choices will have only the intended effects (Weedman, 2008).

System design projects are characterised by uncertainty and ambiguity and are surrounded by various objectives and expectations. System designers must understand and deal with the goal of the design, balance divergent and convergent thinking, handle ill-structured problems that emerge during the process, and move between high levels of abstraction to details (Kimbell, 2009). In the realm of design processes, system designers harness their expertise in architectural and software design, often termed as design logic. The underlying design logic often remains implicit, resulting in a lack of comprehensive guidance on addressing and integrating them within the design process (Kimbell, 2021).

The stakeholders involved each have their 'life worlds' (Weedman, 2008) or logics, something that also needs to be handled. As such, design thinking is an engagement-driven cognitive process, engaging both designers and users (Dell'Era et al., 2020). As Kimbell (2021) observes, assumptions and biases are built into professional practices. These built-in values, practices and accountabilities are rarely examined. However, Micheli et al. (2019, p.130) identified the following designer attributes; abductive reasoning; ability to visualise; blending analysis and intuition; creativity and innovation; gestalt view; interdisciplinary collaboration; iteration and experimentation; Problem-solving; tolerance for ambiguity and failure; and user-centeredness and involvement. Still, they ask for additional research on what constitutes a design thinker and what kind of training and practice is needed to become one.

2.3 The service-dominant perspective in system design

We argue in this paper that integrating UX in agile software design requires taking a socio-technical view in system design projects requires a focus on configuring people's interactions with the system that support work, communication, and decision-making inside as well as across organisational boundaries (Carvalho et al., 2024; Fehrer et al., 2018; Mathiassen and Soreness, 2008). A socio-technical approach delivers systems that are more acceptable to users and provide better value to stakeholders than a purely technical approach (Orlikowski and Iacono, 2001; Baxter and Sommerville, 2011). In line with a socio-technical approach, business value logic becomes vital in a B2B platform development project, since business value logic enables an understanding of a company, its business processes and activities, and competitive factors (e.g., Ojasalo and Ojasalo, 2015). Hence, business model activities need to be designed for and integrated into the specific system (Gatautis, 2017).

Contemporary academic discussions on business value focus on the identification and creation of customer value (cf. Vargo and Lusch, 2011), emphasising value networks and how actors' relationships co-produce value (Blaschke et al., 2019; Jacobides et al., 2006) through a service-dominant logic (SDL) perspective (Lusch and Nambisan, 2015; Vargo

and Lusch, 2011, 2017). SDL broadens the perspective of understanding exchange and value creation where all social and economic actors (e.g., firms, customers) are engaged in exchange to build up a service-providing, value-creating enterprise (Vargo and Lusch, 2011). An SDL perspective involves using the capabilities and skills of an enterprise's actors, i.e., applying resources for the benefit of others or oneself in response to others' needs and desires. In SDL, resources are thought of in a broad sense; they are practically anything, tangible or intangible, an actor can use for support (Lusch and Nambisan, 2015). SDL's view of value is that a company can only offer a value proposition, not deliver value itself, and that this proposition is really an invitation to joint value creation that benefits both parties. By applying an SDL lens, collaborative competences, dynamic capability of customer orientation, and knowledge interfaces that influence innovation outcomes and firm performance, are highlighted (Lusch and Nambisan, 2015). Though, the concept of value is complex, and it can be understood as "the benefit derived by an organisation in proportion to the resources used to achieve it" (Patrício et al., 2023, p.403), but the same benefit has not the same value for all organisations, i.e., one size does not fit all (Patrício et al., 2023).

3 Methodology

In our undertaking to unravel the nuances inherent in the three logics (technological, service-dominant and design) within the agile software design project context, a qualitative approach emerges as a powerful tool to enhance our comprehension of the interwoven complexity. As we navigate through these complexities, our research journey is shaped by an inductive design, harnessing the Gioia methodology (Gioia et al., 2013), also used in other studies with a focus on projects (Civelek et al., 2023; Lagerburg, 2023).

3.1 Research approach and case description

To propel our understanding of the challenges inherent in the design process, as well as the potential of system designers to align diverse logics with distinct situations, we adopted a case study design (Siggelkow, 2007). Our chosen case revolves around a project dedicated to catalysing digitalisation in the manufacturing industry, with a keen focus on designing solutions tailored for businesses across the entire manufacturing value chain. This strategic approach empowered us to identify the nuanced logics within the specific context methodically. Through this methodological lens, we investigated the integration between usability and the agile software design process (cf. Curcio et al., 2019) and unravelled the complex threads of themes and dimensions vital in a system design project from the perspectives of system designers and users.

The authors secured unique access to observe and track the evolution of the design process in the creation of a collaborative B2B-platform from diverse perspectives. As partners in the project, all material created in the agile software design process was open and accessible for the authors to use. This research design enabled us to conceptualise challenges inherent in design and illuminate the strategic pathways for executing agile software design projects, ensuring the delivery of value-added services to end-users. The chosen case serves as a representative instance within the broader many global initiatives

driving digitalisation within the manufacturing industry, shedding light on the hurdles encountered while crafting digital businesses and collaboration.

3.2 Research design and data collection

Agile software design processes present multifaceted work, demanding fine-tuned skills that harmonise the needs of diverse stakeholders. Such choreography ensures that the design process meets and transcends stakeholders' requirements. In our case, a dynamic network of 17 organisations in Europe actively participated and contributed to the agile software design process with different perspectives, e.g., security considerations, legal facets, and user engagement. They contributed with representatives to one or several of the following groups:

1 system designers

2 designers in charge of, e.g., architectural development, security aspects, legal aspects, and user involvement

3 end users operating in the manufacturing industry.

The project partners, from seven countries, had different roles. The partners were organised into three core groups, namely:

1 **The use case group**: each use case consisted of the owner of a problem, i.e., a company in the manufacturing industry, one research partner as a mentor, one software company as a service provider, and one or more supply chain partners. The four use case groups are as follows:

   a *Case White Goods* wanted to improve the flow of information from their customer service and field service back to the internal organisations. The use case focused on third-party SMEs, exemplifying the complex relationship between larger and smaller companies collaborating on a multi-sided platform.

   b *Case Eco Construction* wanted to establish new logistics chains and improve monitoring of transport and on-site construction in order to ensure that reliable quality information on supplier's products from the construction sites was transferred back to the organisation. All stakeholders and possible business partners should be able to publish ordering information (e.g., invoices) and specification documents for direct B2B exchange.

   c *Case Textile Manufacturing* wanted to build close relationships between fabric designers and clothing stylists because their aim was more customised production of their exclusive fabrics. The adoption of virtual prototyping tools would make this tight collaboration possible. The platform should establish a fast and reliable data exchange service, based on IoT, M2M, and B2B data transfer.

   d *Case Wood Furniture* focused on defining and configuring an optimal value chain from a rich and reliable business ecosystem that would make matchmaking between buyers and suppliers possible. Use case vision is to facilitate a collaborative supply partnership and increase innovation capabilities by finding providers of required materials and operations that the use case

cannot cover with its own resources or aims to improve on different levels (i.e., operational costs, ultimate quality).

2 **The core platform developer group** consisted of different IT-system partners and institutes.

3 **The platform adoption group** comprised partners in charge of standardisation and regional industrial sector representatives acting as multipliers for adoption. These partners all had different expertise and competencies ranging from architectural development to user involvement.

The above companies provided input to the system designers. We aimed to crystallise knowledge on agile B2B platform design processes by triangulating data sources involving three main types of data collection:

1 *Participation observation in project meetings and workshops* that encompassed all stakeholder groups, system designers, and end-users. This dynamic occurred within the context of the design project of a B2B-platform. The observations offered insights into the motivations, concepts, requirements, and strategies shaping the design of the B2B-platform and its services. This facilitated understanding of the design process involving, e.g., work plans, and the integration of feedback garnered from ongoing assessments and design trials. In sum, our observations accumulated a total of 68 h observations.

2 *Informal and formal interviews* were carried out with end-users for grasping the design perspectives and reasoning of this group of actors. Nine end-users, encompassing manufacturing companies, suppliers, and customers, were carried out. These respondents were in Sweden and belonged to the construction case. This effort resulted in the accumulation of 17 h of interview data, with individual interviews spanning 45–120 min and an average length of 60 min. The respondents were informed about the purpose of collecting the data and how it was going to be used, hence they were informed and gave their consent.

3 Analysis of written documents regarding the progress of the design process. Specifically internal dialogues between system designers and designers were carried out within a communication platform. Since the authors were part of the project, the communication was accessible, and allowed to be used. This approach was intentionally structured to acquire and continuously amass a substantial dataset revealing insights from the design process. In aggregate, we amassed roughly 150 pages of text material from this communication channel.

3.3 Analyses towards a conceptual framework

Through the utilisation of the Gioia methodology as an approach for data analysis, we were empowered to methodically identify patterns ingrained within the dataset (Gioia et al., 2013). This methodology has been harnessed to discern interconnections both within and between analytical themes by prior researchers (cf. Hafermalz and Riemer, 2021; Seidel et al., 2018). As partners in the project, we early on noticed that in the project agile software design process, all involved had quite different views on what the result should be. As other studies have shown (Edison et al., 2021; Fitriani et al., 2016), we could recognise common challenges, e.g., distributed teams, achieving cross-

functional development team, requirements prioritisation, progress monitoring etc. For instance, requirements prioritisation is complicated, particularly when system design involves multiple stakeholders or customers with multiple competing requirements who have to reach a consensus under limited time and in hectic situations. Further, progress monitoring is manly achieved by project owners and key stakeholders, while feedback from customers and users is only received after they start using the product (Fitriani et al., 2016).

We reasoned that some challenges were due to key actors' divergent logics. Hence, during the initial phase of becoming familiar with the data, we recognised the three logics (design, technological and service-dominant) involved in the agile B2B platform software design process. Furthermore, the open coding approach uncovered a significant distinction in the reliance on various logics among the system designers and users during the agile software design process. This propelled us towards our aim: to explore the technical, the service-dominant, and the design logics involved in the development of a B2B platform, and how these logics interplay in agile software design processes.

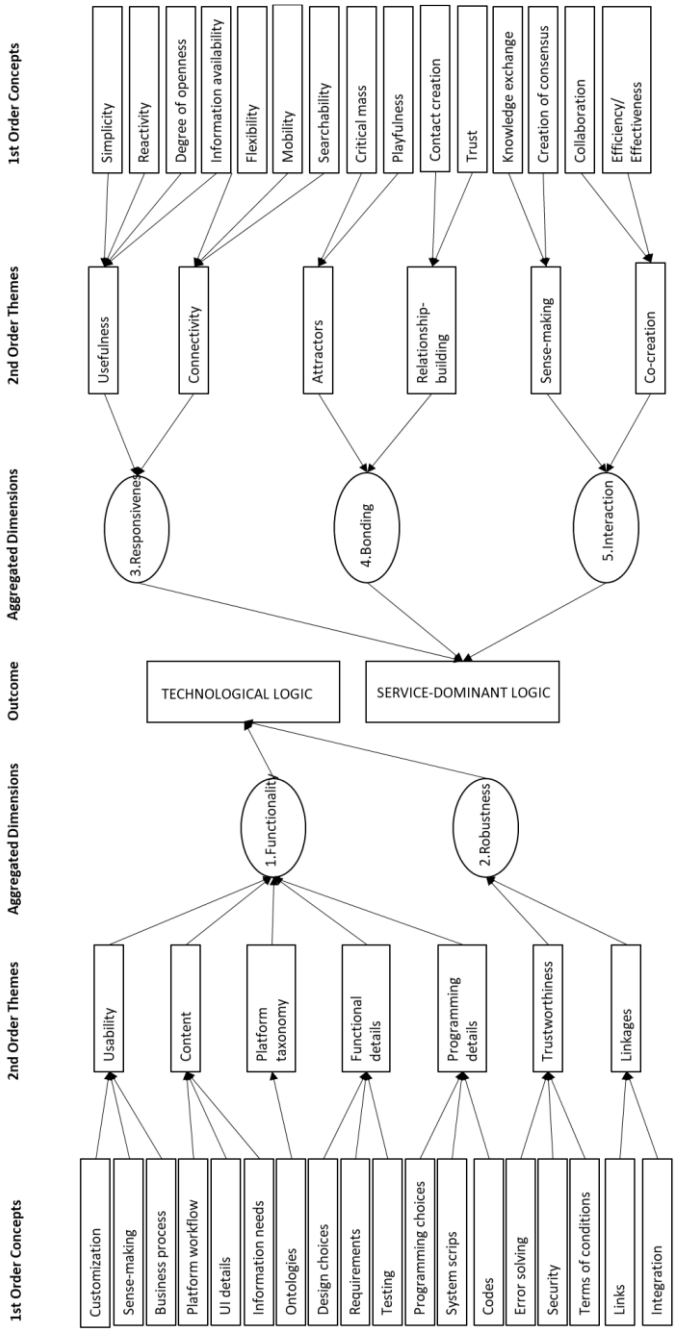
Subsequently, we initiated the coding process, wherein we identified and labelled distinct fragments of information, phrases, and sentences that held significance or meaning. This coding was executed using the transcribed observations, interviews, and downloaded documents as the basis. These analytical undertakings unfolded iteratively, as we constantly cycled between the gathered data and the emerging themes. This iterative approach yielded primary-level codes, secondary-level themes, and overarching dimensions of theoretical conceptualisation. Capturing raw data and representations among the most basic level of categorisation, we identified first-order concepts. We identified 18 concepts categorised as stemming from a design logic, and 15 stemming from a business logic (Figure 1 and Appendix A). This step enabled us to break down data into manageable segments for further analysis.

Hereafter, we aggregated the concepts into second-order themes by looking into relationships among concepts. Through this, we identified seven themes related to design and six themes related to business. This step was hereafter followed by the identification of overarching dimensions. We conducted a detailed analysis of the first-order codes and second-order themes, outlining broader insights that transcend individual themes and offer a more comprehensive understanding of the phenomenon being studied. In this stage, we narrowed down our categorisation into five aggregated dimensions, of which Functionality and Robustness stem from system designers, and Responsiveness, Bonding and Interaction origin from the user perspective. In general, throughout the iterative analysis process, the research team froze some categories, changed others, and removed irrelevant categories. In this way, we captured multiple levels of abstraction, discovering sub-categories and nuances since we successively made our interpretation more theoretical while finding tentative answers to our research questions (cf. Glaser and Strauss, 1967; Shepherd and Williams, 2014).

Drawing on design and business literature, we explored the interplay between the two perspectives. Based on this, we analysed the five overarching dimensions (Functionality, Robustness, Responsiveness, Bonding, and Interaction). Figure 1 illustrates inductive reasoning underpinning the five overarching concepts and theoretical linkage to business and service in the design process. Finally, we identified two logics that the actors in the system design project considered, i.e., the technological and the service-dominant logics. We identified these two logics as important, capturing different stakeholders' needs and wants in the system design project and impacting the success of such projects. Based on

this above, we outlined a conceptual framework for aligning logics in agile software design projects. In this framework, we identified an adaptive space for system design, that involve *sense making*; *designing*; *innovating* and *value creating*, to overcome issues encountered in a system design project and hence, ultimately to create value for the user.

Figure 1 The data structure



To enhance the credibility of the study, a collaborative approach was adopted, involving observations and interviews carried out by the research team, i.e., the three of us. This joint effort not only augmented the potential for reliable outcomes but also facilitated the unearthing of unexpected insights. Thorough field notes were made during both observation sessions and interviews. Moreover, the interviews were recorded and transcribed within a span of 48 h. Then, as part of the data analysis phase, recurring patterns and emerging themes were identified.

4 Outlining un-aligned logics

We identified two aggregated design dimensions and seven second-order themes (see the left side of Figure 1 for the system designers' technological logic and Appendix A, Table A1, for quotations). The system designers' concerns were found among the second-order themes usability; content; platform taxonomy; functional details; and programming details, which we refer to as the design dimension functionality. Also, the system designers put considerable emphasis on designing for trustworthiness, which we refer to as the dimension robustness. Below is a description of the second-order themes.

From an end-user perspective, we identified three aggregated design dimensions, responsiveness, bonding and interaction with six second-order themes (see the right side of Figure 1 and Appendix A, Table A2, for quotations). The design dimension, responsiveness, contains the second-order themes, usefulness and connectivity. The dimension bonding builds on the second-order themes, attractors and relationship building. The third design dimension is interaction containing the second-order themes of sense-making and co-creation. Below follows a description of the second-order themes.

4.1 Design dimensions

Below are the descriptions of the five design dimensions: functionality, robustness, responsiveness, bonding, and interaction. In general, the identified codes were picked since they illustrated a theme that was frequently viewed in the data.

4.1.1 Dimension 1: functionality

Designing for functionality embraced five second-order themes: usability, content, platform taxonomy, functional details, and programming details.

In addressing the first theme, usability, the system designers highlighted three first-order concepts: customisation, sense-making, and business process, where system designers related customisation to the user interface (UI), sense-making to unified understanding and business process to workflows. For the latter, the system designers emphasised the need to test the workflows as a user to grasp the user perspective, e.g., "*When ordering a product, is it possible to have no incoterm? No, I just want to know; what are the requirements from a business perspective?*" "*We are working on an improvement on business processes. Specifically, on the way to refer to the documents* (*messages*) *exchanged in business processes.*".

The second theme, design for content, embraces three first-order concepts: platform workflow, UI details, and information needs. It was important for the system designers to understand what was needed in relation to content for the platform workflows. The UI

was discussed from a technical perspective, focusing om details such as a category's detailed properties, e.g., mapping 'string' to "text". Concerning information, our analysis shows the importance of both being able to access and understand what information is needed to develop adequate content, e.g., "*Products without a price should not be part of our scenario. So, my assumption is that all products have a price…*".

The third theme, design for platform taxonomy, covers the first-order concept sector-specific taxonomy and ontologies, which is important for system designers when classifying and organising information in line with the users' needs.

The fourth theme, design for functional details, comprises three first-order concepts: design choices, requirements, and testing, which are central activities when designing. The system designers expressed the need for valid user requirements and continuous testing to get user feedback.

A fifth and central theme is design for programming details, embracing the first-order concepts programming, scripts and codes. Again, this is at the heart of system designers' work.

4.1.2 Dimension 2: robustness

Designing for robustness involved two second-order themes: trustworthiness and linkages. Hence, the sixth theme, trustworthiness, embraces the three first-order concepts: error solving, security and terms of condition. These concepts are key in developing an attractive platform and, hence, eliminating errors, assuring security around the functions was a main task in platform development, e.g., "*I already digged into the validation process in general because I want to add more security in future releases. There are quite a few approaches that could be included, e.g., VAT validation, SSL certificate lookup, eSignature,…*"

The last theme, linkages, covers two first-order concepts, links and integration. These concepts are also critical for an attractive and user-friendly platform, i.e., the creation of links, and integrating the various functions and features are complex and challenging tasks for system designers.

Thus, in general, the analysis shows that the main emphasis was on technical aspects of the design process, and issues such as user collaboration and business value were absent or rarely addressed.

Our analysis shows that from a user perspective, there is a need to further investigate the **Service-Dominant perspective** since service is the fundamental basis of exchanging in a B2B-platform, value is co-created by multiple actors, and values are uniquely determined by the beneficiary (cf. Vargo et al., 2020). Hence, we investigate how business, collaboration, and co-creation are considered in the design process. Here, we identified three aggregated design dimensions, responsiveness, bonding and interaction, correlating to six second-order themes based on empirical data (see the right side of Figure 1 for the user view and Appendix A, Table A2, for quotations). The design dimension, responsiveness, contains the second-order themes, usefulness, and connectivity. The dimension bonding builds on the second-order themes, attractors and relationship building. The third design dimension is interaction, containing the second-order themes of sense-making and co-creation. Below is a description of the second-order themes.

4.1.3 Dimension 3: responsiveness

Designing for responsiveness embraced two second-order themes: usefulness and connectivity. Designing for usefulness and connectivity is a core endeavour in obtaining user experience quality and, thus, serves the purpose of using the platform. The first second-order theme, usefulness, implies designing for the first-order concepts simplicity, reactivity, degree of openness, and information availability. In short, the users expect a platform that is easy to use, quick to respond, and does not require much effort to get started, e.g., "*If the interface or what you expect from the system is not there, you will not use it. Either it must be very custom-made in advance, or you must give the user the opportunity to dynamically adapt the platform*".

The other second-order theme connectivity involves designing for flexibility, mobility, and searchability. Flexibility encompasses the possibility for users to make different choices, e.g., options concerning tools, channels, and types of data on the platform.

A platform can support the flexibility of obtaining access to different ways of displaying data. Furthermore, the users ask for mobility, i.e., access at anytime and anywhere. The last design concept is searchability, which is central, not least since users nowadays are attuned to, and get good hit rates using the internet.

4.1.4 Dimension 4: bonding

Designing for bonding embraces the two second-order themes: attractors and relationship-building, of which the first, attractors, addresses the need of early adopters and a critical mass of users to up-scale the platform. A high number of users increases the possibility of finding relevant collaboration partners, i.e., making business. Also, playfulness is important for attracting users, and there is a need to promote creativity and encourage playing around and testing different services.

The second-order theme relationship building refers to new contacts and trust. However, when it comes to new contacts, our analysis shows that companies typically prefer to build initial relationships through direct contact and personal meetings. The platform also needs to be designed for trust, which is closely coupled with security and code of conduct. A platform for businesses calls for long-term relationships where mutual trust is developed over time. The contact creation outside the platform assures for some of the trust between the parties on the platform, e.g., "*Code of Conduct is extremely important. When evaluating a supplier, I have to be sure that it is a secure supplier, with good security, high security of delivery with a good economy, good quality and good staff.*"

4.1.5 Dimension 5: interaction

The fifth dimension, designing for interaction, involves the second-order themes: sense-making and co-creation. Sense-making is primarily connected to the first-order concepts of knowledge exchange and the creation of consensus. A platform is expected to contain tools that facilitate sense-making, and possibility to reach consensus e.g., on environmental quality and sustainability. To be noted, there can be a mix of ways to exchange knowledge, such as pure transactions on the platform or through relationships outside the platform. Designing for support can be; "*The platform can facilitate when*

collaboration is already established", and "*We have no such thing as we currently sit in* (*video-conferencing*). *We would like that; it would have been easiest.*"

The second-order theme co-creation relates to the first-order concepts of collaboration and efficiency/effectiveness. Our analysis also showed the need for collaboration around design and production, for example, a fabric supplier can share design or production data with another company. Data changes must then be tracked and made available to all eligible users. To ensure efficiency/effectiveness in co-creation, there is a need for interactive tools that provide quick and accurate data and information exchange. Hence, co-creation can also be about using data from, for example, field technicians by providing unified access to information about a specific product that combines data from different and separate data sources.

4.2 A framework for alignment of logics

Our empirical material shows that system designers, while focusing on their field of expertise (stemming from technological logic) lose sight of the users' wants and needs. Users, on the other hand, approach the design with business value in mind (relating to a service-dominant logic).

The quotations below illustrate the differing perspectives:

> "The user does not understand how difficult it is to develop a rich infrastructure that works properly. The users do not understand our situation. It takes time and resources to design and develop a platform that works properly." (system designer)

> "The system designers do not understand our needs. We can't let our customers in on a platform that is not up and running properly. It takes too long a time to establish a platform that suits my needs." (user)

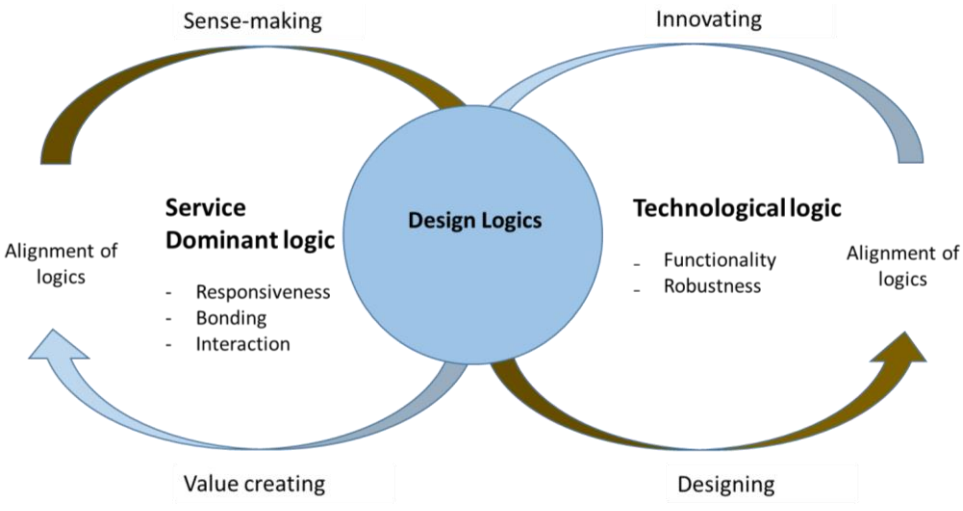
Additionally, the relationship between system designers and users equals a supplier-customer relationship, therefore, our interpretation is that system designers are accountable for aligning the logics throughout the design process.

Our empirical findings from the Gioia analysis apply to four overarching design phases: sense-making, designing, innovating and value creating. These phases were chosen due to the identified dimensions and their content, and since they are important parts of the design process. Sense-making concerns establishing a common view of the design objectives. Designing is about conceptualising the intended solution, and innovating involves transferring the ideas into more concrete services, and finally, value creating, ensures that users' expectations of business value is created. It is important to note that these phases are iterative and bleed into one another, and accordingly, are more refined for each iteration.

The phases sense-making and value creating (see left side Figure 2) stem from service- dominant logic and involves the design dimensions responsiveness, bonding, and interaction (see Figure 1). The phases of designing and innovating are coupled to the technological logic and contain the design dimensions functionality and robustness.

Our interpretation is that the technological and the service-dominant logics need to be aligned for ensuring a holistic perspective permeating the agile software design process. Based on this, we developed a framework for design logics highlighting the need for system designers to align the technological and service-dominant logic into design logics throughout the entire design process.

Figure 2 An adaptive space for agile software design logics (see online version for colours)



At the centre of Figure 2, we introduce Design Logics for integrating and aligning the technological logic and the service-dominant logic, highlighting the view of design as "a situated, contingent set of practices" (Kimbell, 2011). Also, this reinforces the standpoint that system design embraces more than the actual, technical design. Design logics implies the need for continuously integrating different logics, and thus expanding knowledge boundaries, co-creation with users, and commitment (Liedtka et al., 2013).

5 Discussion and conclusions

The framework for adaptive space presented in this paper, addresses the research gap of the importance to integrate UX in Agile software development (cf. Ferreira et al., 2011; Hinderks et al., 2022; Persson et al., 2022). Several scholars point to the importance of designing for communication, interaction, engagement, and the business value of platforms (Brhel et al., 2015; Fitriani et al., 2016; Van Alstyne et al., 2016; Curcio et al., 2019), while several challenges remain. Many of these are related to the human factors, where collaboration and sharing skills are fundamental, and not fully explored (Sampaio et al., 2021). The focus has often been placed on strengthening current capabilities while neglecting new capabilities that could add value (cf. Du et al., 2020). Agile software design projects require an understanding of who the users are, their different roles, what creates value for those different users, and how to achieve commitment (Spagnoletti et al., 2015; Tura et al., 2018). It is necessary to consider underpinning value co-creation processes, and the characteristics of the business environments, as argued by Blaschke et al. (2019).

5.1 The framework for adaptive space

Our results are in line with Sampaio et al. (2021) who highlight the gap between what various users expect from the system designers and what the designers really focus on (or not) besides technical knowledge. Here, our framework for the alignment of logics

addresses the need for a better integration between users' needs and wants, such as prioritisation of nonfunctional usability requirements in comparison to functional requirements (Brhel et al., 2015; Sampaio et al., 2021) to improve the agile software design process (Curcio et al., 2019). System design projects require an understanding of who the actors are, their different roles, what creates value for those different actors, and how to achieve commitment (Spagnoletti et al., 2015; Tura et al., 2018), i.e., it is necessary to consider underpinning value co-creation processes, and the characteristics of the business environments (Blaschke et al., 2019).

Our result shows that in agile software system design projects, the underlying logics of what constitutes value on a B2B-platform becomes non-transparent in the design process, even when different potential actors are involved in the platform design (Tura et al., 2018), and the intention is to design for socio-technical interactions (Spagnoletti et al., 2015). Hence, we identified the need for alignment of such logics in the design process, contributing to systems design theories for design and action (cf. Gregor, 2006). This adds to design in a human- and meaning centred, co-creative, inclusive, holistic, and contextual manner (Karpen et al., 2017), to ensure trust, exchange and development of a value proposition that attracts users (Bonchek and Choudary, 2013; Blaschke et al., 2019; Swart et al., 2022).

We argue that design logics (centre of Figure 2) needs to be encouraged and made transparent in system design projects. Our conceptual framework shows the need for a design logic in the agile software design process (Brown, 2008; Brown and Wyatt, 2010; Dell'Era et al., 2020; Kimbell, 2021; Micheli et al., 2019) and points to the importance of an adaptive space for alignment of logics. We reason that the capability of adopting design logics that aligns service-dominant logic and technological logic is vital. This answers the call for adapting the design to user needs and motivations (Carlsson et al., 2011, Gregor and Hevner, 2013; Hanseth and Lyytinen, 2010). Thus, we provide principles that emphasise the users and their social business context for the systems design process of a B2B platform (Brhel et al., 2015; Sampaio et al., 2021). Our framework, informed by empirical observations, can be used as a tool highlighting dimensions of user-centred agile software design (Pereira and Russo, 2018). The framework, adaptive space for agile software design logics, offers a structured guide for effectively integrating varied logics, with their accompanying dimensions. Grounded in real-world insights, this approach significantly enhances the service and business aspects of agile software design processes, thus increasing the possibility of answering to users' wants and needs within the dynamic context of agile projects.

In sum, this complex context requires structural flexibility in agile software design processes to align the design with user needs and wants. The framework points to a need to 'think through' the design process for the purpose of aligning business, services, and technology to the appropriate context so that the whole service system is optimised. This accords well with findings from previous studies (Karpen et al., 2017; (Brhel et al., 2015; Fitriani et al., 2016; Van Alstyne et al., 2016; Curcio et al., 2019; Sampaio et al., 2021). There is also a need to bridge expectation gaps throughout the adaptive space – making sense, designing and innovating, overcoming the issues encountered and ultimately creating value. We affirm the critical importance of being attentive to the needs of various users.

5.2 Conclusions

The aim of this paper was to explore the technical, service-dominant, and design logics involved in the agile software design process of a B2B platform, and how these logics interplay in agile software design processes.

We have discussed that design logics should combine the service dominant logic, and the technological logic since these logics are fundamental for a B2B platform. We argue that the framework of adaptive space clarifies that different logics need to be aligned in the design process, and hence integrate UX in the Agile software development. The framework highlights the centrality of iteratively addressing user needs and wants. We suggest that the framework can be of guidance since it broadens the focus and makes visible the interplay between logics to be addressed during the phases sense-making, design, innovation, and value creation. In this way, it is possible to move from a technology-driven development to agile software design processes that embrace the full socio-technical context. Design projects benefit from applying principles that support the design of flexibility, relationships, and collaboration.

The conceptual framework of adaptive space is managerially relevant since it provides various dimensions for reflection in design projects. The framework can be used for guidance and support in design processes that strive for heightened awareness of what actors on B2B-platforms consider accessible, trustworthy, and suitable. In conclusion, we argue that adding the service dominant logic would contribute to reaching changes in mind-sets in system design projects. In turn, this would help ensuring acceptance and uptake of B2B-platforms.

While the present study provides a framework of adaptive space for design, several limitations should be acknowledged. Although the study is based on rich data drawn from an agile software design project in the manufacturing context with a B2B focus, our framework needs to be tested in different types of agile software design projects to enable further generalisation. Specifically, it is recommended that more focus should be set on the design process in agile software development and its integration of the different logics. A limitation is also that the framework needs further development emphasising how to make system designers understand the different logics in their development work. Design Thinking techniques and supporting tools used in agile software project development have been widely adopted by professionals in the software industry. However, there are not many practical examples of the technique applied in the literature by using real software projects, and also not many empirical studies that validate the results of using DT in real software systems. As future work, it is intended to investigate and monitor the main challenges faced by practitioners in software development teams when using design thinking, as well as their suggestions for adopting the methodology widely in real contexts.

Acknowledgements

This research has been funded by the European Commission within the H2020 project NIMBLE (Collaborative Network for Industry, Manufacturing, Business and Logistics in Europe), No. 723810, for the period between 01 October 2016–31 September 2019.

References

Agile Manifesto: Manifesto for Agile Software Development (2001) Available at https://agile manifesto.org/ (Accessed 9 September, 2023).

Alqudah, M. and Razali, R. (2016) 'A review of scaling agile methods in large software development', *International Journal on Advanced Science, Engineering and Information Technology*, Vol. 6, No. 6, pp.828–837.

Al-Saqqa, S., Sawalha, S. and AbdelNabi, H. (2020) 'Agile software development: methodologies and trends', *International Journal of Interactive Mobile Technologies*, Vol. 14, No. 11, pp.246–270.

Baxter, G. and Sommerville, I. (2011) 'Socio-technical systems: from design methods to systems engineering', *Interacting with Computers*, Vol. 23, No. 1, pp.4–17.

Blaschke, M., Riss, U., Haki, K. and Aier, S. (2019) 'Design principles for digital value co-creation networks: a service-dominant logic perspective', *Electronic Markets*, Vol. 29, No. 3, pp.443–472.

Bonchek, M. and Choudary, S.P. (2013) 'Three elements of a successful platform strategy', *Harvard Business Review*, Vol. 92, Nos. 1–2, https://hbr.org/2013/01/three-elements-of-a-successful-platform

Bonina, C., Koskinen, K., Eaton, B. and Gawer, A. (2021) 'Digital platforms for development: foundations and research agenda', *Information Systems Journal*, Vol. 31, No. 6, pp.869–902.

Brhel, M., Meth, H., Maedche, A. and Werder, K. (2015) 'Exploring principles of user-centered agile software development: a literature review', *Information and Software Technology*, Vol. 61, pp.163–181.

Brown, T. (2008) 'Design thinking', *Harvard Business Review*, Vol. 86, No. 6, pp.84–92.

Brown, T. and Katz, B. (2011) 'Change by design', *Journal of Product Innovation Management*, Vol. 28, No. 3, pp.381–383.

Brown, T. and Wyatt, J. (2010) 'Design thinking for social innovation', *Development Outreach*, Vol. 12, No. 1, pp.29–43.

Burström, T., Parida, V., Lahti, T. and Wincent, J. (2021) 'AI-enabled business-model innovation and transformation in industrial ecosystems: a framework, model and outline for further research', *Journal of Business Research*, Vol. 127, pp.85–95.

Campanelli, A.S. and Parreiras, F.S. (2015) 'Agile methods tailoring–a systematic literature review', *Journal of Systems and Software*, Vol. 110, pp.85–100.

Canedo, E.D., Pergentino, A.C.D.S., Calazans, A.T.S., Almeida, F.V., Costa, P.H.T. and Lima, F. (2020) 'Design thinking use in agile software projects: Software developers' perception', *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS'20)*, Vol. 978-989-758-423-7. SCITEPRESS, pp.217–224, https://doi.org/10.5220/0009387502170224

Carlsson, S.A., Henningsson, S., Hrastinski, S. and Keller, C. (2011) 'Socio-technical IS design science research: developing design theory for IS integration management', *Information Systems and e-Business Management*, Vol. 9, No. 1, pp.109–131.

Carvalho, A.M., Sampaio, P., Rebentisch, E., Conceição, L., McManus, H., Carvalho, J.Á. and Saraiva, P. (2024) 'In pursuit of the agile organisation: a review and framework development on assessing organisational agility', *International Journal of Agile Systems and Management*, Vol. 17, No. 2, pp.193–220.

Civelek, F., Brem, A., Fritz, K.P. and Zimmermann, A. (2023) 'Product development processes for individualised products: a case study', *IEEE Transactions on Engineering Management*, Vol. 71, pp.6190–6204.

Curcio, K., Santana, R., Reinehr, S. and Malucelli, A. (2019) 'Usability in agile software development: a tertiary study', *Computer Standards and Interfaces*, Vol. 64, pp.61–77.

Dam, R. and Siang, T. (2018) 'What is design thinking and why is it so popular?', *Interaction Design Foundation*, pp.1–6.

de Reuver, M., Sørensen, C. and Basole, R.C. (2018) 'The digital platform: a research agenda', *Journal of Information Technology*, Vol. 33, No. 2, pp.124–135.

Dell'Era, C., Magistretti, S., Cautela, C., Verganti, R. and Zurlo, F. (2020) 'Four kinds of design thinking: from ideating to making, engaging, and criticising', *Creativity and Innovation Management*, Vol. 29, No. 2, pp.324–344.

Du, X., Ji, T. and Hu, Y. (2020) 'How to cooperate in collaborative design: a study of interactive behaviours at different design capability levels', *The Design Journal*, Vol. 23, No. 3, pp.347–369.

Edison, H., Wang, X. and Conboy, K. (2021) 'Comparing methods for large-scale agile software development: a systematic literature review', *IEEE Transactions on Software Engineering*, Vol. 48, No. 8, pp.2709–2731.

Erickson, J., Lyytinen, K. and Siau, K. (2005) 'Agile modeling, agile software development, and extreme programming: the state of research', *Journal of Database Management* (*JDM*), Vol. 16, No. 4, pp.88–100.

Evans, D.S. and Schmalensee, R. (2016) *Matchmakers: The New Economics of Multisided Platforms*, Harvard Business Review Press, Boston.

Fehrer, J.A., Woratschek, H. and Brodie, R.J. (2018) 'A systemic logic for platform business models', *Journal of Service Management*, Vol. 29, No. 1, pp.546–568.

Ferreira, J., Sharp, H. and Robinson, H. (2011) 'User experience design and agile development: managing cooperation through articulation work', *Software Practice Experiment*, Vol. 41, No. 9, pp.963–974.

Fitriani, W.R., Rahayu, P. and Sensuse, D.I. (2016) 'Challenges in agile software development: a systematic literature review', *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pp.155–164, https://doi:10.1109/ICACSIS.2016.7872736.

Fitzgerald, B. (2012) 'Software crisis 2.0', *Computer*, Vol. 45, No. 4, pp.89–91.

Gatautis, R. (2017) 'The rise of the platforms: business model innovation perspectives', *Engineering Economics*, Vol. 28, No. 5, pp.585–591.

Ghayyur, S.A.K., Ahmed, S., Ali, M., Razzaq, A., Ahmed, N. and Naseem, A. (2018) 'A systematic literature review of success factors and barriers of agile software development', *International Journal of Advanced Computer Science and Applications*, Vol. 9, No. 3, pp.278–291.

Gioia, D.A., Corley, K.G. and Hamilton, A.L. (2013) 'Seeking qualitative rigor in inductive research: notes on the Gioia methodology', *Organizational Research Methods*, Vol. 16, No. 1, pp.15–31.

Glaser, B. and Strauss, A. (1967) 'Grounded theory: the discovery of grounded theory', *The Journal of British Sociological Association*, Vol. 12, No. 1, pp.27–49.

Gregor, S. (2006) 'The nature of theory in information systems', *MIS Quarterly*, Vol. 30, No. 3, pp.611–642.

Gregor, S. and Hevner, A.R. (2013) 'Positioning and presenting design science research for maximum impact', *MIS Quarterly*, Vol. 37, No. 2, pp.337–355.

Gustavsson, T. (2023) 'The impact of agile coordination practices on conditions for successful coordination', *International Journal of Agile Systems and Management*, Vol. 16, No. 3, pp.347–367.

Hafermalz, E. and Riemer, K. (2021) 'Productive and connected while working from home: what client-facing remote workers can learn from Telenurses about 'belonging through technology'', *European Journal of Information Systems*, Vol. 30, No. 1, pp.89–99.

Hanseth, O. and Lyytinen, K. (2010) 'Design theory for dynamic complexity in information infrastructures: the case of building internet', *Journal of Information Technology*, Vol. 25, No. 1, pp.1–19.

Hauke-Lopes, A., Ratajczak-Mrozek, M. and Wieczerzycki, M. (2022) 'Value co-creation and co-destruction in the digital transformation of highly traditional companies', *Journal of Business and Industrial Marketing*, Vol. 38, No. 6, pp.1316–1331.

Hein, A., Weking, J., Schreieck, M., Wiesche, M., Böhm, M. and Krcmar, H. (2019) 'Value co-creation practices in business-to-business platform ecosystems', *Electronic Markets*, Vol. 29, pp.503–518.

Hinderks, A., Mayo, F.J.D., Thomaschewski, J. and Escalona, M.J. (2022) 'Approaches to manage the user experience process in agile software development: a systematic literature review', *Information and Software Technology*, Vol. 150, pp.1–15, 106957.

Hoda, R., Salleh, N. and Grundy, J. (2018) 'The rise and evolution of agile software development', *IEEE Software*, Vol. 35, No. 5, pp.58–63.

Hoolohan, C. and Browne, A.L. (2020) 'Design thinking for practice-based intervention: co-producing the change points toolkit to unlock (un) sustainable practices', *Design Studies*, Vol. 67, pp.102–132.

Islind, A.S., Lindroth, T., Lundin, J. and Steineck, G. (2019) 'Co-designing a digital platform with boundary objects: bringing together heterogeneous users in healthcare', *Health and Technology*, Vol. 9, No. 4, pp.425–438.

Jacobides, M.G., Knudsen, T. and Augier, M. (2006) 'Benefiting from innovation: value creation, value appropriation and the role of industry architectures', *Research Policy*, Vol. 35, No. 8, pp.1200–1221.

Jolak, R., Wortmann, A., Liebel, G., Umuhoza, E. and Chaudron, M.R. (2023) 'Design thinking and creativity of colocated vs. globally distributed software developers', *Journal of Software: Evolution and Process*, Vol. 35, No. 5, p.e2377.

Kamalaldin, A., Linde, L., Sjödin, D. and Parida, V. (2020) 'Transforming provider-customer relationships in digital servitisation: a relational view on digitalisation', *Industrial Marketing Management*, Vol. 89, pp.306–325.

Karpen, I.O., Gemser, G. and Calabretta, G. (2017) 'A multilevel consideration of service design conditions: towards a portfolio of organisational capabilities, interactive practices and individual abilities', *Journal of Service Theory and Practice*, Vol. 27, No. 2, pp.384–407.

Kimbell, L. (2009) 'Beyond design thinking: design-as-practice and designs-in-practice', *Conference, CRESC* Manchester, England pp.1–15.

Kimbell, L. (2011) 'Rethinking design thinking: part I', *Design and Culture*, Vol. 3, No. 3, pp.285–306.

Kimbell, L. (2012) 'Rethinking design thinking: part II', *Design and Culture*, Vol. 4, No. 2, pp.129–148.

Kimbell, L. (2021) 'Design as common good', *Online Conference*, 25–26 March, Available at: https://ualresearchonline.arts.ac.uk/id/eprint/16854/1/Kimbell_LogicsofSocialDesign_20210120.pdf (Accessed 7 September, 2023).

Lagerburg, R.A. (2023) *Overcoming Challenges in Software Development Projects: An Integrated Approach for Small and Medium Enterprises (SMEs)*, Master's thesis, University of Twente.

Lehnen, J., Schmidt, T.S. and Herstatt, C. (2016) 'Bringing agile project management into lead user projects', *International Journal of Product Development*, Vol. 21, Nos. 2–3, pp.212–232.

Liedtka, J. (2014) 'Innovative ways companies are using design thinking', *Strategy and Leadership*, Vol. 42, No. 2, pp.40–45.

Liedtka, J., King, A. and Bennett, K. (2013) *Solving Problems with Design Thinking*, Columbia Business School Publishing, New York.

Lusch, R.F. and Nambisan, S. (2015) 'Service innovation: a service-dominant logic perspective', *MIS Quarterly*, Vol. 39, No. 1, pp.155–175.

Mamonov, S. and Peterson, R. (2021) 'The role of IT in organisational innovation–a systematic literature review', *The Journal of Strategic Information Systems*, Vol. 30, No. 4, p.101696.

Marques, R., da Silva, M.M. and Gonçalves, D. (2023) 'Gamification for agile: a systematic literature review', *International Journal of Agile Systems and Management*, Vol. 16, No. 2, pp.226–261.

Mathiassen, L. and Soreness, C. (2008) 'Towards a theory of organisational information services', *Journal of Information Technology*, Vol. 23, No. 4, pp.313–329.

Micheli, P., Wilner, S.J., Bhatti, S.H., Mura, M. and Beverland, M.B. (2019) 'Doing design thinking: conceptual review, synthesis, and research agenda', *Journal of Product Innovation Management*, Vol. 36, No. 2, pp.124–148.

Motwani, J. and Kataria, A. (2023) 'Organisation agility: a literature review and proposed conceptual framework', *International Journal of Agile Systems and Management*, Vol. 16, No. 1, pp.82–123.

Nedeltcheva, G.N. and Shoikova, E. (2017) 'Coupling design thinking, user experience design and agile: towards cooperation framework', *Proceedings of the International Conference on Big Data and Internet of Thing*, pp.225–229, https://doi.org/10.1145/3175684.3175711

Ojasalo, K. and Ojasalo, J. (2015) 'Adapting business model thinking to service logic: an empirical study on developing a service design tool', in Gummerus, J. and von Koskull, C. (Eds.): *The Nordic School – Service Marketing and Management for the Future*, School of Economics, Helsinki, Helsinki, pp.309–333.

Orlikowski, W.J. and Iacono, C.S. (2001) 'Desperately seeking the 'IT' in research–a, IT call to theorising the IT artifact', *Information Systems Research*, Vol. 12, No. 2, pp.121–134.

Paasivaara, M., Behm, B., Lassenius, C. and Hallikainen, M. (2018) 'Large-scale agile transformation at Ericsson: a case study', *Empirical Software Engineering*, Vol. 23, pp.2550–2596.

Palmié, M., Miehé, L., Oghazi, P., Parida, V. and Wincent, J. (2022) 'The evolution of the digital service ecosystem and digital business model innovation in retail: the emergence of meta-ecosystems and the value of physical interactions', *Technological Forecasting and Social Change*, Vol. 177, No. 121496, pp.1–10.

Pande, M. and Bharathi, S.V. (2020) 'Theoretical foundations of design thinking–a constructivism learning approach to design thinking', *Thinking Skills and Creativity*, Vol. 36, No. 100637, pp.1–17.

Parizi, R., Prestes, M., Marczak, S. and Conte, T. (2022) 'How has design thinking being used and integrated into software development activities? A systematic mapping', *Journal of Systems and Software*, Vol. 187, p.111217.

Patrício, T., Pereira, L. and Dias, Á. (2023) 'Benefit realisation management: main barriers and drivers for practice', *International Journal of Agile Systems and Management*, Vol. 16, No. 3, pp.400–428.

Pauli, T., Fielt, E. and Matzner, M. (2021) 'Digital industrial platforms', *Business and Information Systems Engineering*, Vol. 63, pp.181–190.

Pereira, J.C. and Russo, R.F.S.M. (2018) 'Design thinking integrated in agile software development: a systematic literature review', *Procedia Computer Science*, Vol. 138, No. 1, pp.775–782.

Pereira, L., Jerónimo, C., Simões, F., Dias, Á., Costa, R.L.D. and Gonçalves, R. (2024) 'Project virtual teams: systematic literature review', *International Journal of Agile Systems and Management*, Vol. 17, No. 1, pp.15–45.

Pereira, L., Parizi, R., Prestes, M., Marczak, S. and Conte, T. (2021) 'Towards an understanding of benefits and challenges in the use of design thinking in requirements engineering', *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, March, pp.1338–1345, https://doi.org/10.1145/3412841.3442008

Persson, J.S., Bruun, A., Lárusdóttir, M.K. and Nielsen, P.A. (2022) 'Agile software development and UX design: a case study of integration by mutual adjustment', *Information and Software Technology*, Vol. 152, pp.1–10, 107059.

Pidun, U., Reeves, M. and Schüssler, M. (2020) *How do you 'Design' A Business Ecosystem?*, Available at: https://bcghendersoninstitute.com/wp-content/uploads/2020/02/How-Do-You-Design-a-Business-Ecosystem.pdf (Accessed 21 November, 2023).

Piedrahita, C., Alzate-Espinosa, G.A., Herrera-Schlesinger, M.C., Naranjo-Agudelo, A., Araujo-Guerrero, E.F., Torres-Hernández, C.A. and Benítez-Peláez, C.A. (2023) 'Scrum methodology adaptation in the non-software industry: agile management of a research initiative', *International Journal of Agile Systems and Management*, Vol. 16, No. 3, pp.368–399.

Sampaio, S.C., Bastos, T.A. and Marinho, M.L. (2021) 'Soft skills for newborn software engineers in agile teams', *International Journal of Agile Systems and Management*, Vol. 14, No. 1, pp.27–52.

Satria, D., Sensuse, D.I. and Noprisson, H. (2017) 'A systematic literature review of the improved agile software development', *2017 International Conference on Information Technology Systems and Innovation (ICITSI)*, pp.94–99, https://doi:10.1109/ICITSI.2017.8267925

Seidel, S., Chandra Kruse, L., Székely, N., Gau, M. and Stieger, D. (2018) 'Design principles for sensemaking support systems in environmental sustainability transformations', *European Journal of Information Systems*, Vol. 27, No. 2, pp.221–247.

Shepherd, D.A. and Williams, T.A. (2014) 'Local venturing as compassion organising in the aftermath of a natural disaster: the role of localness and community in reducing suffering', *Journal of Management Studies*, Vol. 51, No. 6, pp.952–994.

Siggelkow, N. (2007) 'Persuasion with case studies', *Academy of Management Journal*, Vol. 50, No. 1, pp.20–24.

Spagnoletti, P., Resca, A. and Lee, G. (2015) 'A design theory for digital platforms supporting online communities: a multiple case study', *Journal of Information Technology*, Vol. 30, No. 4, pp.364–380.

Swart, K., Bond-Barnard, T. and Chugh, R. (2022) 'Challenges and critical success factors of digital communication, collaboration and knowledge sharing in project management virtual teams: areview', *International Journal of Information Systems and Project Management*, Vol. 10, No. 4, pp.84–103.

Tessmann, R. and Elbert, R. (2022) 'A multilevel, multi-mode framework for standardisation in digital B2B platform eco-systems in international cargo transportation: a multiple case study', *Electronic Markets*, Vol. 32, No. 4, pp.1843–1875.

Theunissen, T., van Heesch, U. and Avgeriou, P. (2022) 'A mapping study on documentation in continuous software development', *Information and Software Technology*, Vol. 142, p.106733.

Tura, N., Kutvonen, A. and Ritala, P. (2018) 'Platform design framework: conceptualisation and application', *Technology Analysis and Strategic Management*, Vol. 30, No. 8, pp.881–894.

Van Alstyne, M.W., Parker, G.G. and Choudary, S.P. (2016) 'Pipelines, platforms, and the new rules of strategy', *Harvard Business Review*, Vol. 94, No. 4, pp.54–62.

Vargo, S.L. and Lusch, R.F. (2011) 'It's all B2B… and beyond: Toward a systems perspective of the market', *Industrial Marketing Management*, Vol. 40, No. 2, pp.181–187.

Vargo, S.L. and Lusch, R.F. (2017) 'Service-dominant logic 2025', *International Journal of Research in Marketing*, Vol. 34, No. 1, pp.46–67.

Vargo, S.L., Koskela-Huotari, K. and Vink, J. (2020) 'Service-dominant logic: foundations and applications', *The Routledge Handbook of Service Research Insights and Ideas*, Routledge, pp.3–23.

Wangsa, K., Chugh, R., Karim, S. and Sandu, R. (2022) 'A comparative study between design thinking, agile, and design sprint methodologies', *International Journal of Agile Systems and Management*, Vol. 15, No. 2, pp.225–242.

Weedman, J. (2008) 'Client as designer in collaborative design science research projects: What does social science design theory tell us?', *European Journal of Information Systems*, Vol. 17, No. 5, pp.476–488.

Appendix A

Table A1 Interpretive codes, descriptions, and example quotations from a system-developer's and designer's perspective (*D* = designer)

Code	*Description*	*Examples of quotations*
	Second order theme 1: Design aspects for usability	
Customisation	Quotation highlighting design for customisation	"By following the request to hide 'trade details' from the company settings (registration), we should provide customisable UIs for the platform owners, so that they can themselves include/exclude specific features of the platform."
Sense-making	Quotation reflecting on obtaining unified understanding	"D1: YES indeed – in fact, YOU cannot – in isolation – improve the search interface. This is why we are bringing a couple of people on board to strengthen the team."
		D2: Great. Looking forward to getting some unified understanding of improving the platform."
Business process	Quotations indicating design team wanting to explore the workflow process as a user	"When ordering a product, is it possible to have no incoterm? No, I just want to know; what are the requirements from a business perspective?" "We are working on an improvement on business processes. Specifically, on the way to refer to the documents (messages) exchanged in business processes."
	Second order theme 2: Design aspects for content	
Platform workflow	Quotations showing workflow development	"Products without a price should not be part of our scenario. So, my assumption is that all products have a price… (dev 1). In several B2B and B2C platforms prices are on request … so not convinced. It should be the supplier's choice (dev 2)... The logical workflow would be for the user to first start an information request or negotiation."
		"One question about the item information request workflow: should we allow customers to send the request without uploading a technical datasheet to fill? i.e., allow them to just ask a question."
UI details	Quotation giving example of development of UI details	"When looking at category details' properties, we have something like "color STRING". To make it clearer for less technical users, we want to map "STRING" to "TEXT."
		"Quick question for some more control of UI workflow. The `simple search/details` endpoint is dependent on both `catalogued` and `id` parameters, which are obtained from the catalogue. Can there be a possibility where a product may not have these parameters?"
Information need	Quotation showing need for additional information	"I am implementing a service to export catalogues as excel files. However, I need some clarifications to complete this service."

Table A1 Interpretive codes, descriptions, and example quotations from a system-developer's and designer's perspective (*D* = designer)

Code	*Description*	*Examples of quotations*
	Second order theme 3: Platform taxonomy	
Ontologies	Quotations showing work needed on ontologies and taxonomy	"D1: How do I enter data in a schema such as 'children piece of furniture'? Do I need to type in the unit such as 140 cm? D2: Yes, there are such overlaps between the ontologies."
		"Can you please provide me the latest version of the furniture sector ontology?... currently working on a service that consumes the sector specific ontology and provides its structure (product classes and its attributes/properties) with an index."
	Second order theme 4: Design for functional details	
Design choices	Quotations showing different design choices	"That being said, there is the decision to make if 'the platform' wants to make use of the `data-channels` – otherwise deploying a whole Kafka instance just for exchanging a few basic messages is absolutely overkill."
		"I have some issues regarding the design of the product details page and fast track ordering feature, for the record, here are the screens I'd like to discuss."
Requirements	Quotations showing requirement areas to be covered	"There are two requirements to be covered in release 6.0: 1. Exit procedures must be defined (user, organisation).
		2. Platform compliance checks for 3rd party components."
		"I am not sure if we have enough user requirements for this. As a user, I would expect to provide data to properties that are of relevance."
Testing	Quotations highlighting need for user test	"Do you guys have test users with only purchase and only sales permissions?"
		"Registration process much slower than expected. Companies interested; for sure,..., others just waiting document signed as legal representatives."
	Second order theme 5: Design for programming details	
Programming choices	Quotation showing discussion about JavaScript choices	"I wanted to ask about component services in general. I have been using `promises` as service for the 'explorative search component' but wanted to know if I should shift to `observables`. Do you see certain benefits or drawbacks of using observables over promises?"
System scrips	Quotation showing content of script work	"`systemjs.config.js` only has to be used when a library uses some cryptic pathing or depends on certain files. As far as I see, this should not be required for d3."
Codes	Quotation giving examples of code work	"I have found a very odd behaviour with ng-bootstrap for an App I am developing."

Table A1 Interpretive codes, descriptions, and example quotations from a system-developer's and designer's perspective (*D* = designer) (continued)

Code	*Description*	*Examples of quotations*
	Second order theme 6: Design for trustworthiness	
Error solving	Quotations showing fields of errors to be fixed	"I have another error when loading the dashboard: (giving description of the error)."
		"Could you please give some details about the problems so we can take a look? To my observations, fetching multilingual labels of furniture ontology concepts in the runtime seems quite slow. For now, I plan to cache those as soon as the application starts."
Security	Quotation showing development of security issues	"I already dug into the validation process in general because I want to add more security in future releases. There are quite a few approaches that could be included – e.g., VAT validation, SSL certificate lookup, eSignature,..."
Terms of conditions	Quotation indicating terms of conditions to incorporate	"They are set of international commercial terms, but partners can of course define other terms [of conditions]."
	Second order theme 7: Design aspects for linkages	
Links	Quotations showing links to be created	"Privacy compliance (e.g., compliance with the GDPR requirements) – specification of entities with the rights to access the data; user interface comp. with links to privacy policies. Links for users to send privacy related questions."
		"Let's clarify things a bit. It's a fact that some of the services need a means of notifying other services about data changes so those services can update their data accordingly."
Integration	Quotations discussing how to mitigate and integrate	"I am not 100% sure if `search` is the most convenient menu item for it. Are you developing the tracking and tracing as a new component or as part of the explorative search? I suppose the mechanism will be linked to an order business process, right?"
		"Before customised product/service publishing, I'd like to have two major issues dealt with soon: 1) Migration of databases (to improve response times); 2) Integration of multilingually features."

Table A2 Interpretive codes, descriptions, and example quotations from user's perspective

Code	*Description*	*Example*
		Second order theme 1: Usefulness
Simplicity	Quotations expressing a need for simplicity in usability	"If the interface or what you expect from the system is not there, you will not use it. Either it must be very custom-made in advance, or you must give the user the opportunity to dynamically adapt the platform." "While it should be very free, do as you please, there are very different levels of user knowledge. Finding what you are looking for. Otherwise, you will not use it."

Table A2 Interpretive codes, descriptions, and example quotations from user's perspective (continued)

Code	*Description*	*Example*
		Second order theme 1: Usefulness
Reactivity	Quotations showing activities requiring fast reactions	"Valuable, if NN can deliver a solution quickly and get more satisfied customers, the reaction time must be fast, it is important."
		"Have a B2B pilot project on automated ordering, the customer can tick whatever they want at any time (24/7), place an order. It should be simple. Important with availability around the clock."
Degree of openness	Quotations showing why some activities need to be closed	"Open or closed platform... It depends on who uses the platform." "It would be reserved for those with whom we make an agreement. Only contract partners. The more digital information, the more copyable you are, so an agreement is needed at the bottom before being admitted into such an environment."
Information availability	Quotations showing why sharing information is important	"The technology/platform can facilitate when collaboration is established. Find information about products, lots of information about the product content, specifications about the sustainability, the technical information."
		"Obtaining a database of country prescriptions and product requirements would be useful."
	Wood furniture use case	"Awareness of normative… and legislation to enter new markets. A company is interested in entering a new market to export products […] it needs to know all the regulations needed in the destination country."
		Second order theme 2: Connectivity
Flexibility	Quotation indicating requirement of different options	"Both new and existing customers should be able to use the system and order. Mail and telephone are also used for those customers who want to go that route with their orders."
	Textile manufacturing use case	"IoT machine connection and data elaboration. The user shall be able to get access to real time data at machine and product level."
Mobility	Quotations expressing need for interfaces for different types of media	"Every supplier today has some type of presentation on the web, in digital form."
		"Do not walk around with a computer, centrally connecting the mobile to the platform, must be able to read the request on the phone."
	Textile manufacturing	"Managing production from mobile devices."

Table A2 Interpretive codes, descriptions, and example quotations from user's perspective (continued)

Code	*Description*	*Example*
		Second order theme 2: Connectivity
Searchability	Quotations related to ability to search	"If I am looking for a supplier. The easy thing about the internet is that everything is so accessible. I can check everything."
		"Thinking internally, the search function is extremely important […] and that there are rules how to share, how to find it. It must be simple."
	Wood furniture use case	"The manufacturer aims at finding providers of required materials and operations, which it cannot cover by its own resources or which it aims at improving along different dimensions (i.e., operational costs, ultimate quality)."
		"To be able to search for different products or providers is essential […] finding providers of required materials and operations, which it cannot cover by its own resources."
		Second order theme 3: Attractors
Critical mass	Quotation indicating early adoption	"It will be a domino effect for collaboration. A critical mass is needed, from existing suppliers, because then you can create the basis for collaboration and build on it, organically."
Playfulness	Quotation showing need for gamification	"People are used to playing. If our product came into playful context. We have a problem here, we who sell traditional products. We are not visible to the consumer."
		Second order theme 4: Relationship-building
Contact creation	Quotation showing need to design for relationships	"A product catalogue is difficult. A directory is not a person. In our business, there must be a person behind the sale. Works with drawings with customers, together. A good seller can say no to orders."
	Wood furniture use case	"Manufacturer and supplier negotiate all aspects of business conditions."
Trust	Quotation reflecting the need for trustworthiness	"Code of conduct is extremely important. When evaluating a supplier, I must be sure that it is a secure supplier, with good security, high security of delivery with a good economy, good quality and good staff."
		Second order theme 5: Sense-making
Knowledge exchange	Quotations seeking out areas of exchange	"Strength if you could have a platform for collaboration and knowledge exchange. You usually talk about friends' friends (networks, recommendations). Modern networking."
		"There is an interest in our expertise, even outside our country. Selling skills."
Creation of consensus	Quotation indicating need for consensus	"Would like to have a consensus on environmental quality and sustainability."

Table A2 Interpretive codes, descriptions, and example quotations from user's perspective (continued)

Code	*Description*	*Example*
		Second order theme 6: Co-creation
Collaboration	Quotations explaining the need for cooperation	"The platform can facilitate when collaboration is already established."
	White goods use case	"Co-creation can also be about using data from e.g., field technicians: provide unified access to information about a specific product combining data coming from different and separated data sources. Search and data analytics services are expected to accompany this capability."
	Textile manufacturing use case	"The need to be able to collaborate in design and production, e.g., a fabric supplier can share design or production data with another company. Data changes must then be tracked and made available for all eligible users."
Efficiency/ Effectiveness	Quotations seeking out activities that need to be effective	"With an interactive tool, we get more relevant questions from customers, they have come further in the processes. We spend time on the right things today."
		"Being able to discuss a question and not having to answer the same question all the time."
		"Collaboration in finding solutions that save time, less impact on the environment, energy savings on houses, houses air-tight as possible with as little energy as possible."
	Eco construction use case	"The customer of bathrooms can make changes on the features and properties of a bathroom that will be part of the flat in a future eco house. This will be realised by a bathroom product configurator."
	Wood furniture	"Automatic origin certificate declaration. Using NIMBLE, a company can re-collect real-time machine data at the end of the production run and automatically produce the preferential origin certificate declaration."